
Proceedings Paper:
Escolano, Francisco, Curado, Manuel and Hancock, Edwin R. orcid.org/0000-0003-4496-2028 (2016) Commute Times in Dense Graphs. In: Structural, Syntactic, and Statistical Pattern Recognition - Joint IAPR International Workshop, S+SSPR 2016, Mérida, Mexico, November 29 - December 2, 2016, Proceedings. Lecture Notes in Computer Science . , pp. 241-251.

https://doi.org/10.1007/978-3-319-49055-7_22

Commute Times in Dense Graphs

Francisco Escolano, Manuel Curado, and Edwin R. Hancock

Department of Computer Science and AI,
University of Alicante, 03690, Alicante Spain
{sco,mcurado}@dccia.ua.es
Department of Computer Science,
University of York, York, YO10 5DD, UK
erh@cs.york.ac.uk

Abstract. In this paper, we introduce the approach of graph densification as a means of preconditioning spectral clustering. After motivating the need of densification, we review the fundamentals of graph densifiers based on cut similarity and then analyze their associated optimization problems. In our experiments we analyze the implications of densification in the estimation of commute times.

Keywords: Graph densification, Cut similarity, Spectral clustering

1 Introduction

1.1 Motivation

Machine learning methods involving large graphs face a common problem, namely the natural sparsification of data as the number of dimensions d increases. In this regard, obtaining the proximity structure of the data is a key step for the subsequent analysis. This problem has been considered from two complementary perspectives: *efficiency* and *utility*. On the one hand, an efficient, i.e. scalable, proximity structure typically emerges from reducing the $O(dn^2)$ time complexity of kNN graphs, where n is the number of samples. The classical approach for dealing with large graphs is the Nyström method. It consists of sampling either the feature space or the affinity space so that the eigenproblems associated with clustering relaxations become tractable. For instance, in [10] there is a variational version of this method. In [6] an approximated kNN is obtained in $O(dn^t)$ with $t \in (1, 2)$ by recursively dividing and glueing the samples. More recently, anchor graphs [15][13] provide *data-to-anchor* kNN graphs, where $m \ll n$ is a set of representatives (anchors) typically obtained through K-means clustering, in $O(dmnT + dmn)$ where $O(dmnT)$ is due to the T iterations of the K-means process. These graphs tend to make out-of-the-sample predictions compatible with those of Nyström approximations, and in turn their approximated adjacency/affinity matrices are ensured to be positive semidefinite.

On the other hand, the utility of the kNN representation refers to its suitability to predict or infer some properties of the data. These properties include a)

their underlying density and b) the geometry induced by both the shortest path distances and the commute time distances. Concerning the density, it is well known that it can be estimated from the degrees of the kNN graph if its edges contain the local similarity information between the data, i.e. when the graph is weighted. However, when the kNN graph is unweighted the estimation is only acceptable for *reasonably dense* graphs, for instance when $k^{d+2}/(n^2 \log^d n) \to \infty$ as proposed in [20]. However, these densities are unrealistic, since the typical regime, the one adopted in practice, is $k \approx \log n$. A similar conclusion is reached when shortest path distances are analyzed both in weighted and unweighted kNN graphs. The shortest path distance computed from an unweighted kNN graph typically diverges from the geodesic distance. However this is not the case of the one computed from a weighed kNN graph. The solution proposed in [1] consists of assigning proper weights to the edges of the unweighted kNN graphs. Since these weights depend heavily on the ratio $r = (k/(n\mu_d))^{1/d}$, where μ_d is the volume of a $d-$dimensional unit ball, one expects $r \to 0$ for even moderate values of d, meaning that for high dimensional data both unweighted and weighted graphs yield similar, i.e. diverging, estimations. Finally, it is well know that for large $k-$NN (unweighted) graphs the commute time distance can be misleading since it only relies on the local densities (degrees) of the nodes [22][21].

Therefore, for a standard machine learning setting ($n \to \infty$, $k \approx \log n$ and large d) we have that kNN graphs result in a sparse, globally uninformative representation. This can be extended to $\epsilon-$graphs and Gaussian graphs as well. As a result, machine learning algorithms for graph-based embedding, clustering and label propagation tend to produce misleading results unless we are able of preserving the distributional information of the data in the graph-based representation. In this regard, recent experimental results with anchor graphs suggest a way to proceed. In [5], the predictive power of non-parametric regression rooted in the anchors/landmarks ensures a way of constructing very informative weighted kNN graphs. Since anchor graphs are bipartite (only *data-to-anchor* edges exist), this representation bridges the sparsity of the pattern space because a random walk traveling from node u to node v must reach one or more anchors in advance. In other words, for a sufficient number of anchors it is then possible to find links between distant regions of the space. This opens a new perspective for computing meaningful commute distances in large graphs. It is straightforward to check that the spectral properties of the approximate weight matrix $W = Z\Lambda Z^T$, where $\Lambda = \text{diag}(Z^T 1)$ and Z is the data-to-anchor mapping matrix, rely on its low-rank. Then, it is possible to compute a reduced number of eigenvalue-eigenvector pairs associated with a small $m \times m$ matrix, where m is the number of anchors (see [16] for details). In this way, the spectral expression of the commute distance [18] can accomodate these pairs for producing meaningful distances. Our interpretation is that the goodness of the eigenvalue-eigenvector pairs is a consequence of performing kernel PCA process over ZZ^T where the columns of Z act as kernel functions. This interpretation is consistent with the good hashing results obtained with anchor graphs [14][16] where the kernel encoded in the columns of Z is extensively exploited.

Although anchor graphs provide meaningful commute distances with low-complexity spectral representations, some authors have proposed more efficient methods where anchor graphs are bypassed for computing these distances. For instance, Chawla and coworkers [11][9] exploit the fact that commute distances can be approximated by a randomized algorithm in $O(n \log n)$ [19]. Then, using standard kNN graphs with low k for avoiding intra-class noise, their method beats anchor graphs, in terms of clustering accuracy, in several databases. These results are highly contradictory with respect to the von Luxburg and Radl's fundamental bounds (in principle commute distances cannot be properly estimated from large kNN graphs [22]). The authors argue that this can only be explained by the fact that their graphs are quite different from those explored for defining the fundamental bounds (particularly the $\epsilon-$geometric graphs). Their estimator works better than anchor graphs in *dense datasets*, i.e. in settings with a low number of classes and many samples. Our preliminary experiments with the NIST database, with ten classes, confirm that their technique does not improve anchor graphs when data is sparse enough as it happens in a standard machine learning setting.

1.2 Contributions

We claim that one way of providing meaningful estimations of commute distances is to transform the input sparse graph into a *densified graph*. This implies the inference of novel links between data from existing ones. This is exactly what anchor graphs do when incorporate data-to-anchor edges. In this paper, we show that the inference of novel edges can be done by applying recent results in theoretical computer science, namely *cut densification* which in turn is an instance of *graph densification*. Graph densification consists in populating an input graph G with new edges (or weights if G is weighted) so that the output graph H preserves or enforces some structural properties of G. Graph densification offers a principled way of dealing with sparse graphs arising in machine learning so that commute distances can be properly estimated. In this paper we will introduce the main principles of densification and will explore their implications in Pattern Recognition (PR). In our experiments (see the Discussion section) we will show how the associated optimization problems (primal and dual) lead to a reasonable densification (in terms of PR). To the best of our knowledge this is the first application of densification principles to estimate the commute distance.

2 Graph Densification

2.1 Combinatorial Formulation

Graph densification [8] is a principled study of how to significantly increase the number of edges of an input graph G so that the output, H, approximates G with respect to a given test function, for instance whether there exists a given cut. This study is motivated by the fact that certain NP-hard problems have a PTAS

(Polynomial Time Approximation Scheme) when their associated graphs are dense. This is the case of the MAX-CUT problem [2]. Frieze and Kannan [7] raise the question whether this "easyness" is explained by the Szemerédi Regularity Lemma, which states that large dense graphs have many properties of random graphs [12].

For a standard machine learning setting, we have that G is typically sparse either when a kNN representation is used or when a Gaussian graph, usually constructed with a bandwidth parameter t satisfying $t \to 0$, is chosen. Then, the densification of G so that the value of any cut is at most C times the value of the same cut in G is called a *one-sided C-multiplicative cut approximation*. This (normalized) cut approximation must satisfy:

$$\frac{cut_H(S)}{m(H)} \leq C \cdot \frac{cut_G(S)}{m(G)} \ , \tag{1}$$

for any subset $S \subset V$ of the set of vertices V, where $cut_G(S) = \sum_{u \in S, v \in V \sim S} x_{uv}$ considers edge weights $\{x_{uv}\}_{u,v \in V}$ and $x_{uv} \in [0,1]$. For H we have $cut_G(S) = \sum_{u \in S, v \in V \sim S} x'_{uv}$ for edge weights $\{x'_{uv}\}_{u,v \in V}$ also satisfying $x'_{uv} \in [0,1]$. Cuts are normalized by the total edge weight $m(.)$ of each graph, i.e. $m(G) = \sum_{u,v} x_{uv}$ and $m(H) = \sum_{u,v} x'_{uv}$.

Cut Similarity and Optimization Problem. The cut approximation embodies a notion of similarity referred to as *$C-$cut similarity*. Two graphs G and H are C-cut similar if $cut_H(S) \leq C \cdot cut_G(S)$ for all $S \subset V$, i.e. if the sum of the weights in the edges cut is approximately the same in every division. Considering the *normalized version* in Eq. 1, finding the optimal *one-sided $C-$*multiplicative cut densifier can be posed in terms of the following linear program:

$$\begin{aligned} \mathbf{P1} \quad & Max \quad \sum_{u,v} x'_{uv} \\ s.t. \quad & \forall\, u,v: \quad x'_{uv} \leq 1 \\ & \forall\, S \subseteq V: \sum_{u \in S, v \in V \sim S} x'_{uv} \leq C \cdot cut_G(S) \sum_{u,v} x'_{uv} \\ & x'_{uv} \geq 0 \ . \end{aligned} \tag{2}$$

Herein, the term *one-sided* refers only to satisfy the upper bound in Eq. 1.The program **P1** has 2^n constraints, where $n = |V|$, since for every possible cut induced by S, the sum of corresponding edge weights $\sum_{u \in S, v \in V \sim S} x'_{uv}$ is bounded by C times the sum of the weights for the same cut in G. The solution is the set of edge weights x'_{uv} with maximal sum so that the resulting graph H is $C-$cut similar to G. The NP-hardness of this problem can be better understood if we formulate the dual LP. To this end we must consider a *cut metric* $\delta_S(.,.)$ where [4]

$$\delta_S(u,v) = \begin{cases} 1 \text{ if } |\{u,v\} \cap S| = 1 \\ 0 \text{ otherwise} \end{cases} \tag{3}$$

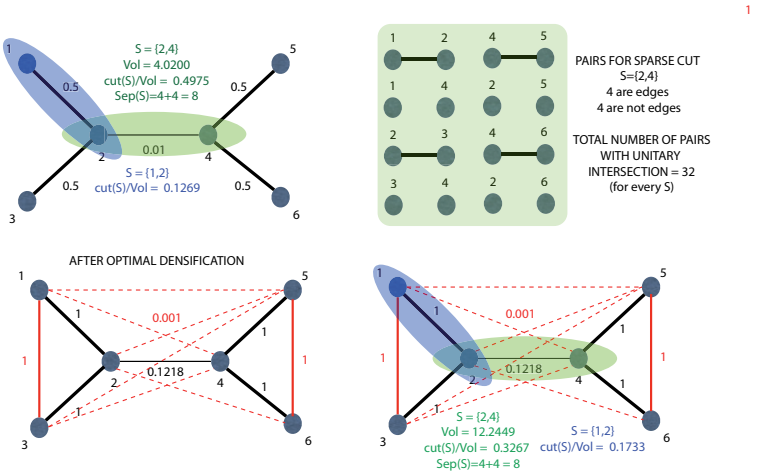

Fig. 1. Densification example. Graph G with $n = 6$ nodes. Top-left: Cuts associated with a couple of sets $S = \{2, 4\}$ and $S = \{1, 2\}$. We define $Sep(S) = \sum_{u,v} \delta_S(u, v)$. For the cut $S = \{2, 4\}$ there are 4 pairs associated with edges and 4 pairs not associated with edges (top-right). This means that this cut is sparse since $\frac{cut(S)}{Vol(G)Sep(S)} = 0.0622$. In bottom-left we show the densification H result solving the spectral version of problem **P1** (Eq. 2) through the dual problem **P2** (Eq. 6) for $C = 2$. Red-dotted lines have weight 0.001. Some cuts have lower values, for instance the one for $S = \{2, 4\}$, whereas others such as the cut for $S = \{1, 2\}$ increase (bottom-right). This is important since the new volume has also increased. All cuts satisfy $\frac{cut_H(S)}{m(H)} \leq C \cdot \frac{cut_G(S)}{m(G)}$.

i.e. δ_S accounts for *pairs of nodes* (not necessarily defining an edge) with an end-point in S. As there are 2^n subsets S of V we can define the following metric ρ on $V \times V$, so that $\rho = \sum_S \lambda_S \delta_S$, with $\lambda_S \geq 0$, is a non-negative combination of a exponential number of cut metrics. For a particular pair $\{u, v\}$ we have that $\rho(u, v) = \sum_S \lambda_S \delta_S(u, v)$ accounts for the number subsets of V where either u or v (but not both) is an end-point. If a graph G has many cuts where $\frac{cut_G(S)/m(G)}{\sum_{u,v} \delta_S(u,v)} \rightarrow 0$ then we have that $\rho(u, v) \geq \mathbb{E}_{(u',v') \in E}\ \rho(u', v')$ since

$$\mathbb{E}_{(u',v') \in E}\ \rho(u', v') = \sum_S \lambda_S \mathbb{E}_{(u',v') \in E}\ \delta_S(u', v') = \sum_S \lambda_S \frac{cut_G(S)}{m(G)}\ . \tag{4}$$

These cuts all called *sparse cuts* since the number of pairs $\{u, v\}$ involved in edges is a small fraction of the overall number of pairs associated with a given subset S, i.e. the graph stretches at a sparse cut. The existence of sparse cuts, more precisely *non-ovelapping sparse cuts* allows the separation of a significant number of vertices $\{u, v\}$ where their distance, for instance $\rho(u, v)$, is larger (to same extent) than the average distance taken over edges. This rationale is posed in [8] as satisfying the condition

$$\sum_{u,v \in V} \min\left\{\rho(u, v) - C \cdot \mathbb{E}_{(u',v') \in E}\ \rho(u', v'), 1\right\} \geq (1 - \alpha)n^2\ , \tag{5}$$

where C is a constant as in the cut approximation, and $\alpha \in (0, 1)$. This means that a quadratic number of non-edge pairs are bounded away from the average length of an edge. In other words, it is then possible to embed the nodes involved in these pairs in such a way that their distances in the embedding do not completely collapse. This defines a so called (C, α) *humble embedding*.
Finding the metric, $\rho(u, v)$ that best defines a humble embedding is the dual problem of **P1**:

$$\begin{aligned} &\mathbf{P2} \quad Min_{\rho = \sum_S \lambda_S \delta_S} \sum_{u,v} \sigma_{uv} \\ &s.t. \qquad \forall\ u, v: \qquad \rho(u, v) - C \cdot \mathbb{E}_{(u',v') \in E}\ \rho(u', v') \geq 1 - \sigma_{uv} \\ &\qquad\qquad\qquad\quad\ \ \sigma_{uv}, \lambda_S \geq 0\ , \end{aligned} \tag{6}$$

where the search space is explicitly the power set of V.
Since the optimal solution of **P2** must satisfy

$$\sigma_{uv} = \max\left\{0, C \cdot \mathbb{E}_{(u',v') \in E}\ \rho(u', v') + 1 - \sigma_{uv}\right\}\ , \tag{7}$$

we have that **P2** can be written in a more compact form:

$$\min_{\rho} \sum_{u,v} \max\left\{0, C \cdot \mathbb{E}_{(u',v') \in E}\ \rho(u', v') + 1 - \sigma_{uv}\right\}\ , \tag{8}$$

which is equivalent to $n^2 - \max_\rho \sum_{u,v} \min\left\{1, \rho(u, v) - C \cdot \mathbb{E}_{(u',v') \in E}\right\}$.
Therefore, a solution satisfying $\sum_{u,v} \sigma_{uv} = \alpha n^2$ implies that the graph has a

humble embedding since

$$\max_{\rho} \sum_{u,v} \min\left\{1, \rho(u,v) - C \cdot \mathbb{E}_{(u',v')\in E}\right\} = (1-\alpha)n^2 \ . \tag{9}$$

Since the σ_{uv} variables in the constraints of **P2** are the dual variables of x_{uv} in **P1**, the existence of a (C, α) humble embedding rules out a C-densifier with an edge weight greater than αn^2 and vice versa.

2.2 Spectral Formulation

Since $Q_G(z) = z^T L_G z = \sum_{e_{uv}\in E} x_{uv}(z_u - z_v)^2$, if z is the characteristic vector of S (1 inside and 0 outside) then Eq. 1 is equivalent to

$$\frac{z^T L_H z}{m(G)} \leq C \cdot \frac{z^T L_G z}{m(G)} \ , \tag{10}$$

for $0-1$ valued vectors z, where L_G and L_H are the respective Laplacians. However, if H satisfies Eq. 10 for any real-valued vector z, then we have a *one-sided C-multiplicative spectral approximation* of G, where L_G and L_H are the Laplacians. This spectral approximation embodies a notion of similarity between the Laplacians L_G and L_H. We say that G and H are *$C-$spectrally similar* if $z^T L_H z \leq C \cdot z^T L_G z$ and it is denoted by $L_H \preceq C \cdot L_G$. Spectrally similar graphs share many algebraic properties [3]. For instance, their effective resistances (rescaled commute times) are similar. This similarity is bounded by C and it leads to nice interlacing properties. We have that the eigenvalues of $\lambda_1, \ldots, \lambda_n$ of L_G and the eigenvalues $\lambda'_1, \ldots, \lambda'_n$ of H satisfy: $\lambda'_i \leq C \cdot \lambda_i$. This implies that H does not necessarily increases the spectral gap of G and the eigenvalues of L_G are not necessarily shifted (i.e. increased).

Whereas the spectral similarity of two graphs can be estimated to precission ϵ in time polynomial in n and $\log(1/\epsilon)$, it is NP-hard to approximately compute the cut similarity of two graphs. This is why existing theoretical advances in the interplay of these two concepts are restricted to *existence theorems* as a means of characterizing graphs. However, the semi-definite programs associated with finding both optimal cut densifiers and, more realistically, optimal spectral densifiers are quite inspirational since they suggest scalable computational methods for graph densification.

Spectral Similarity and Optimization Problem. When posing **P1** and **P2** so that they are *tractable* (i.e. polynomial in n) the cut metric ρ, which has a combinatorial nature, is replaced by a norm in $\mathbb{R}^n$. In this way, the link between the existence of humble embeddings and that of densifiers is more explicit. Then, let $z_1, \ldots, z_n \in \mathbb{R}^n$ the vectors associated with a given embedding. The concept (C, α) humble embedding can be redefined in terms of satisfying:

$$\sum_{u,v\in V} \min\left\{||z_u - z_v||^2 - C \cdot \mathbb{E}_{(u',v')\in E}\ ||z'_u - z'_v||^2, 1\right\} \geq (1-\alpha)n^2 \ , \tag{11}$$

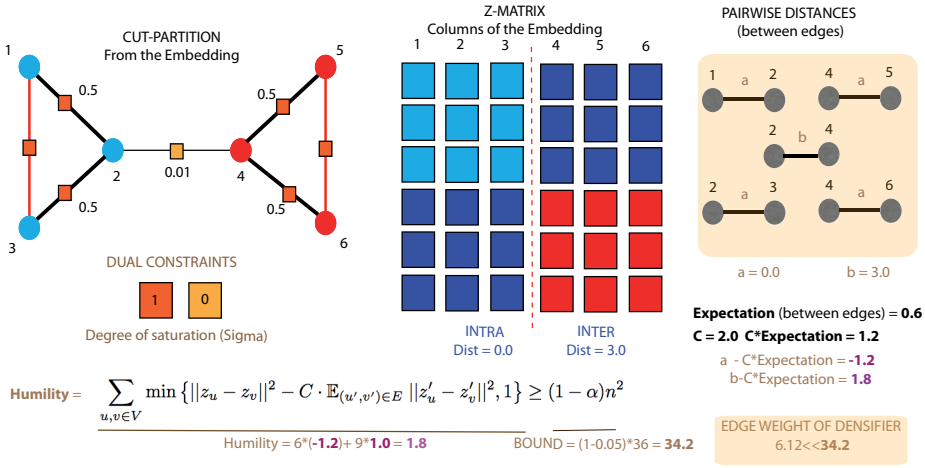

Fig. 2. SDP Dual Solution. Middle: Z-matrix whose columns $z_1, \ldots, z_n$ are the embedding coordinates. Such embedding is optimal insofar it assigns similar coordinates to vertices separated by the sparse cut $S = \{1, 2, 3\}$. Intra-class pairwise distances between columns are close to zero where inter-class distances are close to 3.0. Then the Z matrix encodes the sparse cut itself. Right: to estimate to what extend the columns of Z define a humble embedding, we commence by compute the distances associated with the edges of the graph. This yields $\mathbb{E}_{(u',v')\in E} ||z'_u - z'_v||^2 = 0.6$ where the average is distorted due to the edge $(2, 4)$. Regarding edge pairs, deviations from the expectation are -1.2 for inter-class edges and $+1.8$ for the only inter-class edge. When considering non-edge pairs, for inter-class pairs we have a deviation of $3.0 - 0.6 = 2.4$, whereas for inter-class non-edge pairs, basically $(1, 3)$ and $(5, 6)$ the deviation is negative: -0.6. Therefore, for computing the *humility* of the embedding (see text) we have only 6 deviation smaller than the unit: 4 of these deviations correspond to inter-class edges and 2 of them to intra-class edges. The remainder correspond to 9 non-edge pairs. The resulting humility is 1.8 meaning that $(1 - \alpha)n^2 = 1.8$, i.e. $\alpha = 0.95$. Therefore, the graph has not a one-sided C-multiplicative spectral densifier with edge weight more than $\alpha n^2 = 34.2$. Actually, the weight of the obtained spectral densifier is 6.12. Left: summary of the process in the graph. The colors of the vertices define the grouping given by Z. The colors of the squares indicate whether σ_{uv} are close to 0 (unsaturated constraint) or close to 1 (saturated constraint). Only σ_{24} is unsaturated since $(2, 4)$ distorts the expectation. Variables corresponding two non-edges but linking intra-class vertices are also saturated, namely σ_{13} and σ_{56} (both have a negative deviation). The remaining pairs are unsaturated and they are not plotted for the sake of simplicity.

where distances *between pairs* should not globally collapse when compared with those *between pairs associated with edges*. Then the constraint in **P2** which is associated with the pair $\{u, v\}$ should be rewritten as:

$$||z_u - z_v||^2 - C \cdot \mathbb{E}_{(u',v')\in E} \, ||z'_u - z'_v||^2 \geq 1 - \sigma_{uv} \; . \tag{12}$$

Therefore, **P2** is a linear problem with quadratic constraints. For $Z = [z_1, \ldots, z_n]$ we have that $||z_u - z_v||^2 = b_{uv}^T Z^T Z b_{uv}$ where $b_{uv} = e_u - e_v$. Then, a Semipositive Definite (SPD) relaxation leads to express the first term of the left part of each inequality in terms of $b_{uv}^T Z b_{uv}$ provided that $Z \succeq 0$. Similarly, for the SPD relaxation corresponding to the expectation part of each inequality, we consider the fact that the Laplacian of the graph can be expressed in terms of $L_G = \sum_{u,v} w_{uv} b_{uv} b_{uv}^T$. Since $z^T L_G z = \sum_{(u',v')\in E} w_{uv} ||z(u') - z(v')||^2$, if $z \sim \mathcal{N}(0, Z)$, i.e. z is assumed to be a zero mean vector in $\mathbb{R}^n$ with covariance $Z \succeq 0$, we have that $\mathbb{E}_{(u',v')\in E} \, ||\tilde{z}'_u - \tilde{z}'_v||^2$ can be expressed in terms of $\mathrm{tr}(L_G Z)$ (see [17] for details). Therefore the SDP formulation of **P2** is as follows

$$\begin{aligned} \mathbf{P2_{SDP}} \quad & Min \; \sum_{u,v} \sigma_{uv} \\ s.t. \quad & b_{uv}^T Z b_{uv} - C \cdot \mathrm{tr}(L_G Z) \geq 1 - \sigma_{uv} \\ & Z \succeq 0, \; \sigma_{uv} \geq 0 \; . \end{aligned} \tag{13}$$

Then, the dual problem of $\mathbf{P2_{SDP}}$, i.e. the SDP relaxation of **P1** is

$$\begin{aligned} \mathbf{P1_{SDP}} \quad & Max \; \sum_{u,v} x'_{uv} \\ s.t. \quad & \forall \; u, v : x'_{uv} \leq 1 \\ & \sum_{u,v} x'_{uv} b_{uv} b_{uv}^T \preceq \left(C \cdot \sum_{u,v} x'_{uv} \right) L_G \\ & x'_{uv} \geq 0 \; . \end{aligned} \tag{14}$$

As in the combinatorial version of densification, first we solve the dual and then the primal. The solution of $\mathbf{P2}_{SDP}$ provides σ_{uv} as well as the coordinates of the optimal embedding (in terms of avoiding the collapse of distances) in the columns of Z. In Fig. 2 we explain how the dual solution is obtained for the graph in Fig. 1. We denote the right hand of Eq. 11 as *humility*. The higher the humility the lower the maximum weight of the spectral densifier (as in the combinatorial case).

3 Discussion and Conclusions

With the primal SDP problem $\mathbf{P1}_{SDP}$ at hand we have that $\lambda'_i \leq \left(C \cdot \sum_{u,v} x'_{uv} \right) \lambda_i$ where λ'_i are the eigenvalues of the Laplacian $L_H = \sum_{u,v} x'_{uv} b_{uv} b_{uv}^T$ associated with the densified graph H. For $C > 1$ we have that densification tends to

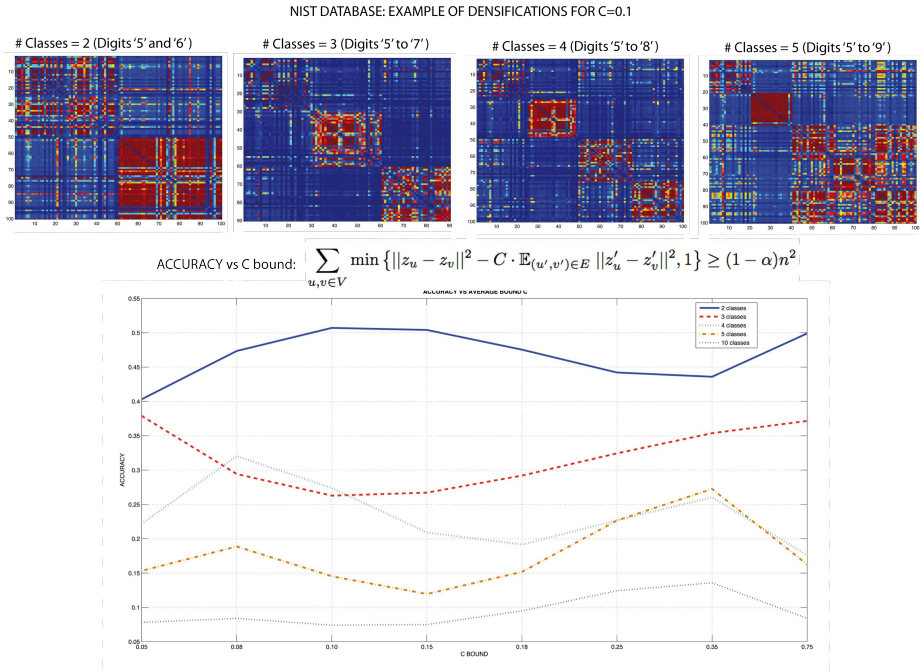

Fig. 3. Densification results for the NIST database.

produce a quasi complete graph $\mathcal{K}_n$. When we add to the cost of the dual problem $\mathbf{P2}_{SDP}$ the term $-K \log\det(Z)$ (a log-barrier) enforces choices for $Z \succeq 0$ (i.e. ellipsoids) with maximal volume which also avoids $\mathcal{K}_n$. In this way, given a fixed $K = 1000$, the structure of the pattern space emerges[1] as we modify the $C < 1$ bound so that the spectral gap is minimized in such a way that reasonable estimations of the commute distance emerge. In Fig. 3 we summarize some experiments done by subsampling the NIST digit database. Given the densifications (more dense in red) the commute time matrix is estimated and the accuracy w.r.t. the ground truth is plotted. Accuracy decreases with the number of classes and in many cases the optimal value is associated with low values of C. The quality of the results is conditioned by the simplicity of the optimization problem (guided only by a *blind* cut similarity, which does not necessarily impose to reduce inter-class noise) but it offers a nice path to explore.

References

1. Alamgir, M., von Luxburg, U.: Shortest path distance in random k-nearest neighbor graphs. In: Proceedings of ICML'12 (2012)

[1] All examples/experiments were obtained with the SDPT3 solver [23] version 4.0. In our experiments, the number of variables is $|E| \approx 4500$ and the SDP solver is polynomial with $|E|$.

2. Arora, S., Karger, D., Karpinski, M.: Polynomial time approximation schemes for dense instances of np-hard problems. Journal of Computer and System Sciences 58(1) 193-210 (1999)
3. Batson, J.D., Spielman, D.A., Srivastava, N., Teng, S.: Spectral sparsification of graphs: theory and algorithms. Commun. ACM 56(8) 87-94 (2013)
4. Benczúr, A.A., Karger, D.R.: Approximating s-t minimum cuts in $O(n^2)$ time. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing 47-55 (1996)
5. Cai, D., Chen, X.: Large scale spectral clustering via landmark-based sparse representation. IEEE Trans. Cybernetics 45(8) 1669-1680 (2015)
6. Chen, J., Fang, H., Saad, Y.: Fast approximate knn graph construction for high dimensional data via recursive lanczos bisection. Journal of Machine Learning Research 10 1989-2012 (2012)
7. Frieze, A.M., Kannan, R.: The regularity lemma and approximation schemes for dense problems. In: 37th Annual Symposium on Foundations of Computer Science, FOCS 96, 12-20 (1996)
8. Hardt, M., Srivastava, N., Tulsiani, M.: Graph densification. In: Innovations in Theoretical Computer Science 2012, 380-392 (2012)
9. Khoa, N.L.D., Chawla, S.: Large scale spectral clustering using approximate commute time embedding. CoRR abs/1111.4541 (2011)
10. Vladymyrov, M., Carreira-Perpinan, M.A.: The Variational Nystrom method for large-cale spectral problems. ICML'16 211-220 (2016)
11. Khoa, N.L.D., Chawla, S.: Large Scale Spectral Clustering Using Resistance Distance and Spielman-Teng Solvers. In: Ganascia, Jean-Gabriel, Lenca, Philippe and Petit, Jean-Marc (Eds): DS 2012, LNCS vol. 7569, 7-21 (2012)
12. Komlós, J., Shokoufandeh, A., Simonovits, M., Szemerédi, E.: The regularity lemma and its applications in graph theory. In: Theoretical Aspects of Computer Science, Advanced Lectures 84-112 (2000)
13. Liu, W., He, J., Chang, S.: Large graph construction for scalable semi-supervised learning. In: Proceedings of ICML'10 679 - 686 (2010)
14. Liu, W., Mu, C., Kumar, S., Chang, S.: Discrete graph hashing. In: NIPS'14, 3419-3427 (2014)
15. Liu, W., Wang, J., Chang, S.: Robust and scalable graph-based semisupervised learning. Proceedings of the IEEE 100(9) 2624-2638 (2012)
16. Liu, W., Wang, J., Kumar, S., Chang, S.: Hashing with graphs. In: Proceedings of ICML'11, 1-8 (2011)
17. 16. Luo, Z., Ma, W., So, A.M., Ye, Y., Zhang, S.: Semidefinite relaxation of quadratic optimization problems. IEEE Signal Processing Magazine 27(3) 20-34 (2010)
18. Qiu, H., Hancock, E.R.: Clustering and embedding using commute times. IEEE TPAMI 29(11) 1873-1890 (2007)
19. Spielman, D.A., Srivastava, N.: Graph sparsification by effective resistances. SIAM J. Comput. 40(6) 1913-1926 (2011)
20. von Luxburg, U., Alamgir, M.: Density estimation from unweighted k-nearest neighbor graphs: a roadmap. In: NIPS'13 225-233 (2013)
21. von Luxburg, U., Radl, A., Hein, M.: Getting lost in space: Large sample analysis of the resistance distance. In: NIPS'10 2622-2630 (2010)
22. von Luxburg, U., Radl, A., Hein, M.: Hitting and commute times in large random neighborhood graphs. Journal of Machine Learning Research 15(1) 1751-1798 (2014)
23. Toh, K.C., Todd M., Tutuncu, R.: SDPT3 - A MATLAB software package for semidefinite programming. Optimization methods and Software 11 545-581 (1998)